

# TOTALVIEW

## RELEASE NOTES



MAY 2003  
VERSION 6.2.0-3

Copyright © 1999–2003 by Etnus LLC. All rights reserved.

Copyright © 1998–1999 by Etnus Inc. All rights reserved.

Copyright © 1996–1998 by Dolphin Interconnect Solutions, Inc.

Copyright © 1993–1996 by BBN Systems and Technologies, a division of BBN Corporation.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise without the prior written permission of Etnus Inc (Etnus).

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Etnus has prepared this manual for the exclusive use of its customers, personnel, and licensees. The information in this manual is subject to change without notice, and should not be construed as a commitment by Etnus. Etnus assumes no responsibility for any errors that appear in this document.

TotalView and Etnus are registered trademarks of Etnus Inc. TimeScan and Gist are trademarks of Etnus Inc.

All other brand names are the trademarks of their respective holders.



# Contents

## TotalView Issues

<b>TotalView News</b> .....	2
Group > Reload symbols command no longer necessary .....	2
Using libdbfork on AIX.....	2
.tvdrc File .....	2
X Resources and preferences .....	3
cset demangler option replaced by xlf .....	3
<b>Problems Fixed</b> .....	3
Problems Fixed On All Platforms .....	3
Action Point > At Location: Typing C++ member function .....	4
Action Point > At Location: No longer torn down .....	4
Ambiguous functions from dynamically loaded shared libraries are found when looking up a function .....	5
Bulk launch sometimes failed if single debug server launch was disabled ..	5
Buttons now responsive if two process windows are open .....	5
C++ class variables were incorrectly marked as 'Stale' .....	5
Diving on an Fortran 90 variable can cause a fatal error (IRIX and Linux) ..	5
Error Message Problem: resolve_final_symbol error.....	5
Fortran modules list item order.....	5
GCC 3.2 g77: General note.....	6
Group > Delete problem fixed.....	6
Keyboard accelerator notation standardized.....	6
Launch strings now include %B by default.....	6
Loading a breakpoint file for a shared library could crash TotalView .....	6
Message Queue Graph node placement .....	6
-pid command line option added.....	6
Program Browser: Cannot be used until process starts .....	6

Search Paths: Fatal Error starting up with very large search path values....	7
Setting breakpoints in functions located in two different libraries problem fixed.....	7
Spell checker problems resolved.....	7
Threads not proceeding.....	7
tvpane.cxx crash .....	7
UPC shared addresses of shared variables and shared pointers given as local .....	7
Variable names in Variable Window title .....	7
-verbosity command line option no takes precedence.....	8
View > Lookup Function/File problem .....	8
Window focus changes affected some users .....	8
HP Alpha Tru64 Problems Fixed .....	8
C++ static variables not visible.....	9
"exAssert in file tvpanes.cxx line 1190" problem fixed .....	9
Files can now be found through symbolic links .....	9
GCC 3.1: TotalView can parse debug information For stl::deque.....	9
KCC 4.0F: Problem parsing information from C++ class solved.....	9
KCC: Startup problem resolved .....	9
libpthread.so problem resolved.....	9
Loading executable problem fixed .....	9
Memory Usage command now shows correct values of heap usage .....	9
Optimizing compiler variables no longer show misleading values .....	10
prun bulk launch no longer crashes tvdsr.....	10
Reading symbols for pointer problem fixed .....	10
Reading symbols causing seg fault problem fixed.....	10
Reloading breakpoint file problem fixed.....	10
IBM AIX Problems Fixed .....	10
2 GB code size problem fixed .....	11
Cray pointers issue .....	11
Diving on a Fortran 90 local variable problem fixed.....	11
Form_long result too large problem fixed .....	11
Fortran logical type issue.....	11
Fortran parameter lookup problem fixed .....	11
GCC 3.2: Internal Error starting up upon a GCC 3.2 compiled executable.....	11
Large data segments no longer crash TotalView .....	11
MPI processes randomly stop before hitting a user breakpoint during startup .....	12
Multi-dimensioned const arrays no longer crashes TotalView.....	12

Multi-threaded core debugging problem fixed.....	12
Linux IA-64 Problems Fixed .....	12
Fortran static link problem .....	12
MPT Support.....	12
Linux x86 Problems Fixed .....	13
Attaching problem fixed .....	13
CodeRoad JNI Bridge attaching problem fixed.....	13
configure_autostart licensing problem fixed.....	13
File > Preferences > Formatting: Defaults Command No Longer Crashes TotalView .....	14
GCC -gstabs compiler option support is improved .....	14
GCC 3.1 and 3.2: Opaque Type Problem When Base Class in a Different Shared Library Problem Resolved .....	14
Lahey/Fujitsu Fortran: Assembly shown by mistake .....	14
Lahey/Fujitsu Fortran: Showing wrong source .....	14
Lahey/Fujitsu Fortran: Stepping problems fixed .....	14
Miscounting processors .....	14
printf() problems.....	14
RMS prun and Message Queue .....	14
Watchpoint warning message suppressed .....	15
SGI IRIX Problems Fixed .....	15
Breakpoints no longer disappear when using group step, next, or run to .....	15
Fortran Modules Support .....	15
Incorrect statistics and visualization of UPC distributed arrays .....	15
Loading a breakpoint file for a shared library could crash TotalView .....	15
STL type transformations limitation.....	16
TotalView crash problem fixed .....	16
Value of remote elements of shared arrays not correctly represented.....	16
Sun SPARC Solaris Problems Fixed .....	16
"Attempt to get an address from a located symbol ..." problem fixed .....	16
CodeRoad JNI Bridge: Apparent hanging problem fixed .....	17
"Current scope not a block" error fixed .....	17
DBX class tag errors.....	17
"Duplicate entry in table" error .....	17
Fortran 90 breakpoint problem fixed .....	17
Missing library load notifications when using JNI Bridge problem fixed ...	17
MPI and Sun ClusterTools 5: Focus problem .....	17
Sun Cluster Tools 5 MPI problem fixed .....	17
Sun WorkShop 5.0: Local variables .....	18
TotalView terminates with an internal error and no other information ....	18

<b>Known Problems</b>	18
Problems on All Platforms	18
C++ exceptions	19
C++ not supported in the evaluation system	20
dprint does not display the contents of a common block or module.	21
Evaluation system: casting return values	21
Evaluation system: fortran intrinsics such as WRITE and COS not supported	21
Evaluation System: Spaces required and case sensitive	21
Evaluation point with a goto and a step	22
Fortran 90 modules not supported in the evaluation system	22
Fortran arrays whose size changes	22
Function static variables may be invisible when using KCC	22
GUI: Accelerators are overriding menu mnemonics if F10 is pressed	23
Multithreaded corefile display problem	23
Portland Group Compilers (or PGI)	23
PVM	23
Type transformations system revised; older version is incompatible	23
Variables in subblocks may not be found when using GCC 3.2	23
Watchpoints not allowed on registers	23
Xoftware and Motif problems	24
HP Alpha Tru64 Problems	24
Anonymous unions using GNU	24
Compiling with -C to detect subscripts	25
Fortran 90 modules not in stack frame in HP (Compaq) Fortran	25
Opaque type showing up on Tru64	25
Planting too many action points causes problems	25
Pointers may show incorrect values using GCC 3.1.1 on Tru64 4.0f	26
prun problem finding executable file during MPI debugging	26
Setting a breakpoint in a large shared-memory target causes a SEGV	26
Thread debugging problems on all versions of HP Tru64	26
HP Alpha Linux Problems	27
Fortran 90 modules problem	27
IBM AIX RS/6000 Problems	27
AIX may only create a partial core file	28
Array statistics and visualization problems when using very large arrays	28
Calling dynamic objects from Evaluation Window	29
Certain XLF compiler options may show incorrect addresses for common blocks in 64-bit mode	29

Continuing from a breakpoint in a 64-bit multithreaded application may cause the application to fail .....	29
Multithreaded problems .....	29
poe interferes with a standalone CLI's use of stdin .....	30
ptrace attaching fails .....	30
Process contention scope not supported .....	30
pthdb_pthread() returns an empty pthread list .....	31
Signals are not delivered to the thread the user requested .....	31
Watchpoints are not supported on the Power4 architecture .....	31
XL Fortran: Compiler omits information about modules in OpenMP programs .....	32
XL Fortran: Not all versions supported .....	32
xlF 8.1.0.0 compiler emits broken module debug information .....	32
Linux ia64 Problems .....	32
SGI's MPT not supported .....	32
Linux x86 Problems .....	32
Breakpoints in C++ constructors in shared libraries problem .....	33
Calling exec() from a thread problems .....	33
GCC g77 problem with common blocks .....	33
GLIBC update required for RedHat 7.3 .....	33
Intel Fortran: 128-bit real support issue .....	33
Licensing problem using Intel Hyperthreading Processors .....	33
Multithreaded corefiles on Linux are not understood .....	34
Native Posix Thread Library (NPTL) not yet supported .....	34
Opening message queue on some versions of Quadrics/RMS crashes TotalView .....	34
PGI Problems .....	34
Red Hat Linux kernel 2.4 security update causes seg fault .....	34
Red Hat 9 not yet supported .....	36
Stepping into system routine backtrace problem .....	36
Thread debugging and errno .....	37
SGI IRIX Problems .....	37
Arrays in main are not found unless declared in common .....	37
Cray pointers in common blocks broken .....	38
Evaluation system forces real function result into a long temporary .....	38
#include and -cpp Used Together in Fortran 90 .....	38
KCC does not put original file name into symbol table .....	39
main not found by TotalView .....	39
Values in assumed sized arrays may be wrong for F77 compiled routines .....	39

Sun SPARC Solaris Problems .....	39
Apogee 4.0 compilers must be patched.....	39
Breakpoints in thunks may cause crash .....	39
SUNPERF runs slow when debugging a Fortran 90 program .....	40
<b>Reporting Problems</b> .....	<b>40</b>
Notices .....	40





## TotalView Issues

These Release Notes are for TotalView version Version 6.2.0-3. It contain important information that affects your software.

This document contains information about bugs that existed in previous versions that have been fixed and bugs that still exist.

Other booklets you may need to read are:

- "TotalView Platforms": Lists the platforms and environments in which TotalView runs.
- "Special IBM Considerations": Describes special considerations when running TotalView on IBM platforms.
- "Special Linux Considerations": Describes special considerations when running TotalView on Linux platforms.
- "Patching": Outlines the procedures for installing patches to software provided by other vendors that must be done before you can TotalView.

You can obtain these documents in the following ways:

- PDF and HTML versions can be obtained from <http://www.etnus.com/Support/docs>.
- PDF and HTML versions are contained with the documentation tar file that you can download when you are downloading TotalView from our web site.
- A HyperHelp version of this file can be viewed after installing TotalView by selecting the **Help > Release Notes** command.

The manuals for this release are *TotalView Users Guide*, *TotalView Reference Guide*, and *TotalView Installation Guide*. Their version number is 6.2.

## TotalView News

Consult the “New Features” section within TotalView’s Help system for a discussion of the new features contained within TotalView. Our web site highlights these changes.

This section describes changes that are not mentioned in that document. New items are in bold.

- Group > Reload symbols command no longer necessary
- Using libdbfork on AIX
- .tvdrc File
- X Resources and preferences
- cset demangler option replaced by xlf

### *Group > Reload symbols command no longer necessary*

TotalView will automatically reload symbols if it discovers you have recompiled your program

### *Using libdbfork on AIX*

On AIX, 64-bit binaries for version 4.3 are completely different from 64-bit binaries for version 5.1. Thus, you will need to link with a version of the dbfork library that is specific to your operating system. If you are creating 64-bit code on AIX 4.3, use:

```
/usr/totalview/lib.libdbfork_64.a \  
-bkeepfile:/usr/totalview/lib.libdbfork_64.a
```

On AIX 5.1 use:

```
/usr/totalview/lib.libdbfork_64_51.a \  
-bkeepfile:/usr/totalview/lib.libdbfork_64_51.a
```

### *.tvdrc File*

The location of the **.tvdrc** has moved. Beginning at Version 6.0, TotalView creates a **.totalview** subdirectory within your home directory. It expects to find your **.tvdrc** within this subdirectory. Temporarily, TotalView will also look in your home directory for a **.tvdrc** if it doesn’t find a **.tvdrc** file in the **.totalview** subdirectory. This means that you do not need to move your startup file immediately. However, we do not guarantee that future versions will look in both places.

### *X Resources and preferences*

Prior to Version 6.0, you could alter TotalView's behavior by setting values in an **.Xdefaults** file. Beginning at Version 6.0, you should either use TotalView's preferences or set variables in a **.tvdrc** file. TotalView will read your **.Xdefaults** file the first time it comes up in version 6 and then translate any values it finds into variables, and then write these variables into its preferences file. After this time, it will ignore your **.Xdefaults** file.

If, however, you are placing entries in an **.Xdefaults** file that modify the Visualizer's behavior, these entries will continue to be used.

### *cset demangler option replaced by xlf*

The **cset** demangler option to the **-demangler** command line option and the **TV::current\_cplus\_demangler** variable is deprecated and replaced by the **xlf** option.

## Problems Fixed

The following sections list the problems that have been fixed.

- Problems Fixed On All Platforms on page 3
- HP Alpha Tru64 Problems Fixed on page 8
- IBM AIX Problems Fixed on page 10
- Linux IA-64 Problems Fixed on page 12
- Linux x86 Problems Fixed on page 13
- SGI IRIX Problems Fixed on page 15
- Sun SPARC Solaris Problems Fixed on page 16

### Problems Fixed On All Platforms

The following problems have been fixed. More details about each item follow this list. Items new with this release are in bold.

- Action Point > At Location: Typing C++ member function (6.1.0-2)
- Action Point > At Location: No longer torn down (6.1.0-2)
- Ambiguous functions from dynamically loaded shared libraries are found when looking up a function (6.1.0-2)

- Bulk launch sometimes failed if single debug server launch was disabled (6.2.0-0)
- Buttons now responsive if two process windows are open (6.1.0-3)
- C++ class variables were incorrectly marked as 'Stale' (6.2.0-0)
- Diving on an Fortran 90 variable can cause a fatal error (IRIX and Linux) (6.2.0-0)
- Error Message Problem: resolve\_final\_symbol error (6.1.0-2)
- **Fortran modules list item order** (6.2.0-3)
- GCC 3.2 g77: General note (6.1.0-2)
- Group > Delete problem fixed (6.1.0-3)
- Keyboard accelerator notation standardized (6.2.0-0)
- Launch strings now include %B by default (6.2.0-0)
- Loading a breakpoint file for a shared library could crash TotalView (6.1.0-3)
- Message Queue Graph node placement (6.1.0-2)
- -pid command line option added (6.2.0-0)
- **Program Browser: Cannot be used until process starts** (6.2.0-3)
- Search Paths: Fatal Error starting up with very large search path values (6.1.0-2)
- Setting breakpoints in functions located in two different libraries problem fixed (6.2.0-0)
- Spell checker problems resolved (6.2.0-0)
- Threads not proceeding (6.1.0-3)
- tvpane.cxx crash (6.2.0-0)
- UPC shared addresses of shared variables and shared pointers given as local (6.1.0-2)
- Variable names in Variable Window title (6.1.0-2)
- -verbosity command line option no takes precedence (6.1.0-2)
- View > Lookup Function/File problem (6.1.0-2)
- Window focus changes affected some users (6.2.0-0)

**Action Point > At Location: Typing C++ member function**

Problem trying to set a breakpoint by typing the C++ member function name into this dialog box was resolved (6.1.0-2: 4518)

**Action Point > At Location: No longer torn down**

TotalView no longer tears down this dialog window when a function name is misspelled (4510: 6.1.0-2)

### *Ambiguous functions from dynamically loaded shared libraries are found when looking up a function*

When looking up a function defined in more than one dynamically loaded shared libraries, TotalView will now present an Ambiguous Function dialog box that lets you pick which version of the function should be opened. (4488: 6.1.0-2)

### *Bulk launch sometimes failed if single debug server launch was disabled*

If the single debug server launch was disabled by using TotalView's Preferences Dialog Box and the bulk launch only required one server to be started, TotalView would fail with an internal error. While this problem is fixed, you should be aware that even with a bulk launch, TotalView may actually be using a single server launch. (4673: 6.2.0-0)

### *Buttons now responsive if two process windows are open*

In addition, closing the first window no longer causes an internal error. (4600: 6.1.0-3)

### *C++ class variables were incorrectly marked as 'Stale'*

Diving on certain class variables brought up a Variable Window that indicated the values were **Stale**, even following lines where they had just been updated. (4567: 6.2.0-0)

### *Diving on an Fortran 90 variable can cause a fatal error (IRIX and Linux)*

This problem occurred either when diving on a local variable that is being referenced inside a contained subroutine or when diving on a module that contained only data. The error you saw began as follows:

```
Fatal error: index 8 out of range [0, 8) in
st_nameset_set_t::get_current_interpretation
...
(4538: 6.2.0-0)
```

### *Error Message Problem: resolve\_final\_symbol error*

TotalView no longer generates thousands of **ERROR: Couldn't resolve\_final\_symbol** has been addressed (4545: 6.1.0-2)

### *Fortran modules list item order*

The modules listed by the **Tools > Fortran Modules** command are no longer alphabetical. (4800: 6.2.0-3)

### ***GCC 3.2 g77: General note***

Several known issues debugging applications compiled with this compiler when using TotalView 6.0 were resolved (4583: 6.1.0-2)

### ***Group > Delete problem fixed***

Entering the **Group > Delete** command on a multi-threaded or multi-process program sometimes crashed TotalView. The following message was displayed.

Fatal error: Can't live without a process  
(4609)

### ***Keyboard accelerator notation standardized***

The notation that indicated keyboard accelerators within TotalView's menus has been regularized. (4686: 6.2.0-0)

### ***Launch strings now include %B by default***

The default **tvdsrv** and Visualizer launch strings now use the **%B** replacement character. The **%B** expands to the full path to the **bin** directory of the running TotalView installation. (4491, 4629: 6.2.0-0)

### ***Loading a breakpoint file for a shared library could crash TotalView***

When this happened, TotalView displayed the following error message:

Fatal error: tv\_target\_addrlist\_t access out of range Terminated  
(4501: 6.1.0-1)

### ***Message Queue Graph node placement***

Node placement is now preserved after clicking the **Update** button (4403: 6.1.0-2)

### ***-pid command line option added***

The new **-pid** command line option lets you specify a running process as a debug target. (2312: 6.2.0-0)

### ***Program Browser: Cannot be used until process starts***

You cannot use the Program Browser until the target process starts. This fixes a problem where the wrong variable data could be shown or TotalView might crash when diving on these variables. (4809: 6.2.0-3)

### *Search Paths: Fatal Error starting up with very large search path values*

Problems that caused TotalView to terminate with errors like **Fatal Error: Attempt to allocate a temp string containing X bytes** or **TotalView: FATAL ERROR STARTING UP: form\_long result too large** when you entered an especially long search path are resolved. (3462: 6.1.0-2)

### *Setting breakpoints in functions located in two different libraries problem fixed*

Setting a breakpoint on a function located in two different libraries or images after a *dlopen* no longer crashes TotalView when you click **Go** and the *dlopen* attaches to the shared library. The error message displayed was:

Fatal error: db\_breakpoint\_t::register\_sourcelevel\_blocks: Unexpected number of blocks in one target (4616: 6.2.0-0)

### *Spell checker problems resolved*

The spell checker that existed in version 5 no longer exists. Instead, it has been integrated with TotalView's ambiguous function system. This is documented in the TOTALVIEW USERS GUIDE. (4505: 6.2.0-0)

### *Threads not proceeding*

Threads did not proceed off of a "When hit, stop thread" breakpoint when you entered a **Group->Go** or **Group->Next** command. (4144, 4145, 4660, 4666: 6.1.0-3)

### *tvpane.cxx crash*

In some cases, if you had modified the source code after an executable was compiled, an internal error in TotalView could occur due to inconsistent line number information in the executable. (4560: 6.2.0-0)

### *UPC shared addresses of shared variables and shared pointers given as local*

When debugging a UPC application, TotalView was correctly displaying the value of objects of type pointer to shared. However, it did not correctly representing the shared addresses of objects that were themselves shared. (4508: 6.1.0-2)

### *Variable names in Variable Window title*

These titles are no longer corrupted after several dives (4463: 6.1.0-2)

***-verbosity command line option no takes precedence***

The **-verbosity** command line option now take precedence over any use of the **dset VERBOSE** variable in **.tvdrc** files. (4530: 6.1.0-2)

***View > Lookup Function/File problem***

Using this command to open a header file no longer causes a crash (4514: 6.1.0-2)

***Window focus changes affected some users***

Window focus behavior changed at version TotalView 6. If window focus follows the pointer rather than by clicking in a window, you saw changes when new windows were raised. For example, clicking in the Root Window to change the current process would change focus to the Process Window. Clicking again on a different process in the Root Window was disabled until focus was brought back to the Root Window. Totalview 6.2 now only changes the focus for the 'find' window. (4549: 6.2.0-0)

**HP Alpha Tru64 Problems Fixed**

The following problems have been fixed. More details about each item follow this list.

- **C++ static variables not visible** (6.2.0-3)
- "exAssert in file tvpanes.cxx line 1190" problem fixed (6.2.0-0)
- Files can now be found through symbolic links (6.2.0-0)
- GCC 3.1: TotalView can parse debug information For stl::deque (6.0.0-1)
- KCC 4.0F: Problem parsing information from C++ class solved (6.0.0-1)
- **KCC: Startup problem resolved** (6.2.0-3)
- **libpthread.so problem resolved** (6.2.0-3)
- Loading executable problem fixed (6.1.0-3)
- Memory Usage command now shows correct values of heap usage (6.0.0-1)
- Optimizing compiler variables no longer show misleading values (6.0.0-1)
- **prun bulk launch no longer crashes tvdsrv** (6.2.0-3)
- Reading symbols for pointer problem fixed (6.2.0-0)
- Reading symbols causing seg fault problem fixed (6.2.0-0)
- Reloading breakpoint file problem fixed (6.2.0-0)



### ***C++ static variables not visible***

C++ static variables were not visible either when diving or when using the **View > Lookup Variable** command. (4827: 6.2.0-3)

### ***"exAssert in file tvpanes.cxx line 1190" problem fixed***

TotalView sometimes issued the non-fatal error "exAssert in file tvpanes.cxx line 1190" when undiving in the Source Pane. (4665: 6.2.0-0)

### ***Files can now be found through symbolic links***

TotalView was sometimes unable to locate source files when debugging programs through symbolic links. The search path rules have been improved. See Setting Search Paths in Chapter 3 of the TOTALVIEW USERS GUIDE. (4683: 6.2.0-0)

### ***GCC 3.1: TotalView can parse debug information For stl::deque***

When using the GCC 3.1 compiler on Tru64 with the STL container **stl::deque**, TotalView crashed. This no longer occurs. (4014: 6.0.0-1)

### ***KCC 4.0f: Problem parsing information from C++ class solved***

Previously, TotalView would crash, producing the following error message when it processed debugging information for certain types of C++ classes compiled with the KCC 4.0f compiler: (4482: 6.0.0-1)

Fatal error: Cannot convert from st\_error\_type\_t to st\_aggregate\_type\_t

### ***KCC: Startup problem resolved***

TotalView no longer crashes when it starts up upon certain KCC compiled codes. (4793: 6.2.0-3)

### ***libpthread.so problem resolved***

A problem where you could receive a TotalView error when you looked up variables in **libpthread.so** is resolved. (4808: 6.2.0-3)

### ***Loading executable problem fixed***

TotalView no longer crashes with a segmentation fault while loading an executable. (4649: 6.1.0-3)

### ***Memory Usage command now shows correct values of heap usage***

TotalView no longer displays the following error message after you invoked the **Tools > Memory Usage** command on Tru64:

ERROR: ifb\_relocate\_address: Bogus address ...

The heap usage should now be correct. (4511: 6.0.0-1)

### *Optimizing compiler variables no longer show misleading values*

If you used the Compaq compiler's **-g3 -fast** command-line options, TotalView would show values for some variables that the optimizing compiler had removed. (4437: 6.0.0-1)

### *prun bulk launch no longer crashes tvdsrv*

A prun bulk launch no longer causes the **tvdsrv** to get a segmentation violation. (4777: 6.2.0-3)

### *Reading symbols for pointer problem fixed*

Internal errors resulting from reading the symbol for a pointer to a derived type defined in an Fortran 90 module are resolved. (4699, 4706: 6.2.0-0)

### *Reading symbols causing seg fault problem fixed*

A segmentation fault that occurred when TotalView read symbols on some specific executables no longer occurs. (4649: 6.2.0-0)

### *Reloading breakpoint file problem fixed*

The fatal error that occurred after reloading a breakpoint file containing breakpoints in *dlopened* library is fixed. (4501: 6.2.0-0)

## IBM AIX Problems Fixed

The following problems have been fixed. More details about each item follow this list. Items new with this release are in bold.

- 2 GB code size problem fixed (6.2.0-0)
- Cray pointers issue (6.1.0-2)
- Diving on a Fortran 90 local variable problem fixed (6.1.0-3)
- Form\_long result too large problem fixed (6.2.0-0)
- **Fortran logical type issue** (6.2.0-2)
- Fortran parameter lookup problem fixed (6.2.0-0)
- GCC 3.2: Internal Error starting up upon a GCC 3.2 compiled executable (6.1.0-2)
- Large data segments no longer crash TotalView (6.1.0-3)

- MPI processes randomly stop before hitting a user breakpoint during startup (6.1.0-3)
- Multi-dimensioned const arrays no longer crashes TotalView (6.1.0-3)
- **Multi-threaded core debugging problem fixed** (6.2.0-3)

### ***2 GB code size problem fixed***

The error debugging code with data size greater than 2 GB is resolved. When this problem occurred, TotalView displayed the following error message:

```
TotalView: FATAL ERROR STARTING UP: Attempt to register overlapping
asects ...
(4646: 6.2.0-0)
```

### ***Cray pointers issue***

Problem diving on cray pointers in common blocks when using xlf 7.1 was resolved (4521: 6.1.0-2)

### ***Diving on a Fortran 90 local variable problem fixed***

Diving on a local variable in a Fortran 90 module would sometimes fail, causing a 'not found' dialog box to pop up. This occurred when the **Prefered Scope** field showed incorrect information regarding the current scope. (4617: 6.1.0-3)

### ***Form\_long result too large problem fixed***

A problem with the C++ demangler would sometimes result in variable names that were too long to fit into an internal buffer. (4601: 6.2.0-0)

### ***Fortran logical type issue***

TotalView now displays Fortran logical types correctly (4708: 6.2.0-2)

### ***Fortran parameter lookup problem fixed***

The TotalView internal error resulting from attempting to lookup a Fortran parameter has been resolved. (4632: 6.2.0-0)

### ***GCC 3.2: Internal Error starting up upon a GCC 3.2 compiled executable***

A TotalView crash starting up upon an executable compiled with GCC 3.2 was fixed. The GCC 3.2 compiler failed to emit correct debug information, which caused an internal error within TotalView. (4507: 6.1.0-2)

### ***Large data segments no longer crash TotalView***

The error messages you saw were (in part):

TotalView: FATAL ERROR STARTING UP: read\_all\_symbols:  
After save-and-stop, the target still isn't stopped

and

TotalView: FATAL ERROR STARTING UP: Attempt to register  
overlapping asects (4524, 4646: 6.1.0-3)

### *MPI processes randomly stop before hitting a user breakpoint during startup*

MPI processes randomly stopped before hitting a user breakpoint during startup of an MPI job. We are aware that this may occur under some circumstances and are working on an improved fix. (4634: 6.1.0-3)

### *Multi-dimensional const arrays no longer crashes TotalView*

This occurred when you examined these kinds arrays in C programs compiled with gcc 2.95 (4544: 6.1.03)

### *Multi-threaded core debugging problem fixed*

TotalView no longer crashes with the following message when it is debugging a multi-threaded core file: (4807: 6.2.0-3)

Fatal error: While trying to workaround a GNU V3 compiler bug,  
we failed to find a subroutine at provisional address ...

## Linux IA-64 Problems Fixed

The following problems have been fixed. More details about each item follow this list. Items new with this release are in bold.

- **Fortran static link problem** (6.2.0-3)
- **MPT Support** (6.2.0-3)

### *Fortran static link problem*

TotalView no longer produces an internal error when you are debugging a statically linked Fortran program. (4817: 6.2.0-3)

### *MPT Support*

MPT is now supported on SGI's ALTIX series of IA-64 Linux servers. The supported version of MPT is 1.7. (4761: 6.2.0-3)

## Linux x86 Problems Fixed

The following problems have been fixed. More details about each item follow this list. Items new with this release are in bold.

- Attaching problem fixed (6.1.0-3)
- **CodeRoad JNI Bridge attaching problem fixed** (6.2.0-3)
- `configure_autostart` licensing problem fixed (6.1.0-3)
- File > Preferences > Formatting: Defaults Command No Longer Crashes TotalView (6.0.0-1)
- GCC `-gstabs` compiler option support is improved (6.1.0-3)
- GCC 3.1 and 3.2: Opaque Type Problem When Base Class in a Different Shared Library Problem Resolved (6.0.0-1)
- Lahey/Fujitsu Fortran: Assembly shown by mistake (6.1.0-2)
- Lahey/Fujitsu Fortran: Showing wrong source (6.1.0-2)
- **Lahey/Fujitsu Fortran: Stepping problems fixed** (6.2.0-3)
- Miscounting processors (6.2.0-0)
- `printf()` problems (6.2.0-0)
- **RMS `prun` and Message Queue** (6.2.0-3)
- Watchpoint warning message suppressed (6.1.0-3)

### *Attaching problem fixed*

Previously, attaching to a running process could cause the following fatal error:

```
TotalView: FATAL ERROR STARTING UP: LID for the main executable is null
```

This occurred if you had recompiled the process while a copy was still active or if the `proc` file system had mistakenly marked it as deleted. (4656: 6.1.0-3)

### *CodeRoad JNI Bridge attaching problem fixed*

The CodeRoad JNI Bridge no longer fails to attach to the Java Virtual Machine when running under NetBeans. This occurred when NetBeans was installed in a deep directory tree. (4741: 6.2.0-3)

### *configure\_autostart licensing problem fixed*

The **`configure_autostart`** installation script no longer fails to set up the necessary hooks to start the license manager daemon in the commonly-used runlevel 5 (graphical log in). (4611: 6.1.0-3)

*File > Preferences > Formatting: Defaults Command No Longer Crashes TotalView*  
(4473)

***GCC -gstabs compiler option support is improved***

Support for executables compiled with the GNU GCC compiler with the **-gstabs** compilation option has been improved. (4088, 4089)

***GCC 3.1 and 3.2: Opaque Type Problem When Base Class in a Different Shared Library Problem Resolved***

Sometimes, when diving into a class whose base class was in a different shared library, TotalView would show the base class as an opaque type. (4466)

***Lahey/Fujitsu Fortran: Assembly shown by mistake***

Previously, assembly code was displayed instead of the source file when diving on F90 modules, even when the module has executable statements. (4374: 6.1.0-2)

***Lahey/Fujitsu Fortran: Showing wrong source***

Previously, TotalView could show source or assembly code for the wrong file when you dove or used **View > Lookup Function** on a Fortran 90 module. This occurred when diving on modules that contained only data (that is, it did not have any executable statements) (4158: 6.1.0-2)

***Lahey/Fujitsu Fortran: Stepping problems fixed***

Problems with stepping certain Lahey/Fujitsu Fortran 95 programs no longer occur. (3658: 6.2.0-3)

***Miscounting processors***

TotalView was miscounting processors on Hyperthreading Xeons running Red Hat 8.0. (4564: 6.2.0-0)

***printf() problems***

On Red Hat 7.3 with either the default packages or the updated i686 packages, calling **printf()** in an **eval point** crashed the target. (4203: 6.2.0-0)

***RMS prun and Message Queue***

Error messages related to and problems displaying the message queue when debugging RMS prun programs no longer occur. (4790: 6.2.0-3)

### *Watchpoint warning message suppressed*

TotalView no longer displays the following message when you set watchpoints on long long values:

Attempt to add wrong child type to a homogeneous RowColumn widget  
(4546)

## SGI IRIX Problems Fixed

The following problems have been fixed. More details about each item follow this list. Items new with this release are in bold.

- Breakpoints no longer disappear when using group step, next, or run to (6.0.0-1)
- **Fortran Modules Support** (6.2.0-3)
- Incorrect statistics and visualization of UPC distributed arrays (6.1.0-2)
- Loading a breakpoint file for a shared library could crash TotalView (6.2.0-0)
- STL type transformations limitation (6.2.0-0)
- **TotalView crash problem fixed** (6.2.0-3)
- Value of remote elements of shared arrays not correctly represented (6.1.0-2)

### *Breakpoints no longer disappear when using group step, next, or run to*

When issuing a group level next, step, or run to command, breakpoints on the lines that were stepped over could disappear. (4469: 6.0.0-1)

### *Fortran Modules Support*

The Fortran Modules window now shows Fortran 90 modules for programs compiled using the MIPSpro compilers. (3798: 6.2.0-3)

### *Incorrect statistics and visualization of UPC distributed arrays*

TotalView presented incorrect data to the Visualizer when asked to display array data that was stored in UPC arrays of type **shared**. TotalView was also unable to correctly generate summary statistics on such an array. These problems are resolved. (4484: 6.1.0-2)

### *Loading a breakpoint file for a shared library could crash TotalView*

When this happened, TotalView displayed the following error message:

Fatal error: tv\_target\_addrlist\_t access out of range Terminated  
(4501: 6.1.0-1)

***STL type transformations limitation***

The new STL type transformations were designed for MIPSpro C++ 7.3.1.3m.  
There are known problems with older MIPSpro compilers. (4750: 6.2.0-0)

***TotalView crash problem fixed***

TotalView no longer crashes when processing debugging information. The error message displayed states that it happened while processing the dwarf frame table.  
(3798: 6.2.0-3)

***Value of remote elements of shared arrays not correctly represented***

A synchronization problem that prevented TotalView from displaying the correct values for remote elements of a UPC **shared** array when stopped during an initialization loop was resolved. (4500: 6.1.0-2)

**Sun SPARC Solaris Problems Fixed**

The following problems have been fixed. More details about each item follow this list. Items new with this release are in bold.

- "Attempt to get an address from a located symbol ..." problem fixed (6.2.0-0)
- **CodeRoad JNI Bridge: Apparent hanging problem fixed** (6.2.0-3)
- "Current scope not a block" error fixed (6.2.0-0)
- DBX class tag errors (6.1.0-3)
- "Duplicate entry in table" error (6.2.0-0)
- Fortran 90 breakpoint problem fixed (6.1.0-3)
- MPI and Sun ClusterTools 5: Focus problem (6.1.0-2)
- Sun Cluster Tools 5 MPI problem fixed (6.1.0-3)
- Sun WorkShop 5.0: Local variables (6.1.0-2)
- TotalView terminates with an internal error and no other information (6.1.0-3)

***"Attempt to get an address from a located symbol ..." problem fixed***

The following fatal error no longer occurs on some GCC 3.X executables "Attempt to get an address from a located symbol that does not directly address memory".  
(4680: 6.2.0-0)



### *CodeRoad JNI Bridge: Apparent hanging problem fixed*

The JVM within some versions of the JDK used the SEG signal during startup. As TotalView catches this signal, the JVM was stopped, giving the appearance that the Bridge and the Java IDE were hung. (4781: 6.2.0-3)

### *"Current scope not a block" error fixed*

If a Fortran 90 program used modules, TotalView would sometimes crash with the following error message:

```
Fatal error: Current scope is null or is not a block ...  
(4569: 6.2.0-0)
```

### *DBX class tag errors*

TotalView no longer generates a series of error messages relating to a DBX class tag. (4623: 6.1.0-3)

### *"Duplicate entry in table" error*

TotalView would terminate with a "Duplicate entry in table" fatal error when a **sleep()** was placed between **mpi\_finalize()** and a **return**. (4009: 6.2.0-0)

### *Fortran 90 breakpoint problem fixed*

The problem that caused TotalView to fail when you set a breakpoint in an Fortran 90 no longer occurs. This problem was associated with the following error message:

```
Fatal error: Current scope is null or is not a block (4569: 6.1.0-3)
```

### *Missing library load notifications when using JNI Bridge problem fixed*

(4460).

### *MPI and Sun ClusterTools 5: Focus problem*

When debugging an MPI application using Sun ClusterTools 5, TotalView now focuses on **main()** after the ranked processes are stopped for setting breakpoints (4267: 6.1.0-2)

### *Sun Cluster Tools 5 MPI problem fixed*

When running some Sun Cluster Tools 5 MPI programs through without stopping, TotalView would sometimes crash and display the following error message:

```
Fatal error: Kill returned code -1 when sending a stop signal  
Terminated (4556: 6.1.0-3)
```

***Sun WorkShop 5.0: Local variables***

Problem with showing local variables in Fortran 90 program compiled with Sun WorkShop 5.0 resolved (4519: 6.1.0-2).

***TotalView terminates with an internal error and no other information***

(4621: 6.1.0-3)

## Known Problems

The following sections list the problems that have been found.

- Problems on All Platforms on page 18
- HP Alpha Tru64 Problems on page 24
- HP Alpha Linux Problems on page 27
- IBM AIX RS/6000 Problems on page 27
- Linux ia64 Problems on page 32
- Linux x86 Problems on page 32
- SGI IRIX Problems on page 37
- Sun SPARC Solaris Problems on page 39

You may find your problem (and its solution) documented on our website's FAQ, which is located at <http://www.etnus.com/Support/faqs.html>.

## Problems on All Platforms

The following are problems that we know about. More details about each item follow this list.

- C++ exceptions
- C++ not supported in the evaluation system (6.0.0-0)
- dprint does not display the contents of a common block or module. (6.0.0-0)
- Evaluation system: casting return values (6.0.0-0)
- Evaluation system: fortran intrinsics such as WRITE and COS not supported
- Evaluation System: Spaces required and case sensitive
- Evaluation point with a goto and a step
- Fortran 90 modules not supported in the evaluation system (6.0.0-0)

- Fortran arrays whose size changes
- Function static variables may be invisible when using KCC
- GUI: Accelerators are overriding menu mnemonics if F10 is pressed (5.0.0-0)
- Multithreaded corefile display problem (5.0.0-0)
- Portland Group Compilers (or PGI) (6.0.0-0)
- PVM (6.0.0-0)
- Type transformations system revised; older version is incompatible (6.0.0-0)
- Variables in subblocks may not be found when using GCC 3.2 (6.0.0-0)
- Watchpoints not allowed on registers (5.0.0-0)
- Xoftware and Motif problems (5.0.0-0)

### C++ exceptions

TotalView does not have full support for C++ exceptions. Single-stepping over code that will throw an exception is problematic and often results in the process running away. To help with this situation, TotalView will detect when an exception throw is going to occur while single-stepping.

By default, TotalView brings up a dialog box to ask if you wish to stop the process. Answering **No** continues the process. Be aware that if you are stepping within the “try” block, your process may run away. Answering **Yes** stops the process upon entry into the system runtime routine that issues the throw. This is a temporary solution and full C++ exception handling may be provided in a future TotalView version.

This mechanism is available for all supported C++ compilers on the supported platforms for SGI IRIX 6.x, Power AIX, Alpha HP Tru64, and Sun SPARC Solaris platforms.

You can change this behavior if you select the **Warn about C++ exceptions during single step operations** checkbox within the **File > Preferences** dialog box’s Options Page. This option lets you toggle the warning on and off.

If this option is turned off, TotalView does not catch C++ exception throws during single-step operations. This may cause the single-step operation to lose control on the process and cause it to run away.

### *C++ not supported in the evaluation system*

C++ code fragments are not yet supported by TotalView. You will need to use C wrapper code if you need to use call methods in the evaluation system. Here is an example that uses a C wrapper that lets you call these methods.

```
#include <stdio.h>

class Foo {

public:

    Foo() { a=0; b=0; c=0;}

    void set_a(int rhs) { a = rhs;}
    void set_b(int rhs) { b = rhs;}
    void set_c(int rhs) { c = rhs;}

    int get_a() { return a;}
    int get_b() { return b;}
    int get_c() { return c;}

    void operator=(Foo &rhs) {
        a = rhs.a;
        b = rhs.b;
        c = rhs.c;
    }

private:

    int a;
    int b;
    int c;

};

extern "C" {
    int foo_get_a(Foo &tmp) {
        return tmp.get_a();
    }

    int foo_get_b(Foo &tmp) {
        return tmp.get_b();
    }

    int foo_get_c(Foo &tmp) {
        return tmp.get_c();
    }

    void foo_set_a(Foo &tmp, int new_a) {
        tmp.set_a(new_a);
    }

    void foo_set_b(Foo &tmp, int new_b) {
        tmp.set_b(new_b);
    }

    void foo_set_c(Foo &tmp, int new_c) {
        tmp.set_c(new_c);
    }

    void foo_equals(Foo &lhs, Foo &rhs) {
        lhs = rhs;
    }
}
```

```

    }
    }

using namespace std;

int main (int argc, char *argv[]) {

    Foo bar1;
    Foo bar2;

    bar1.set_a(10);
    bar1.set_b(20);
    bar1.set_c(30);

    bar2 = bar1;

    return 0;
}

```

This lets you call one of **bar1**'s or **bar2**'s C++ methods in the expression evaluation system using a code fragment like the following:

```
foo_set_a(&bar1, 1024);
```

(6.0.0-0)

### *dprint does not display the contents of a common block or module.*

You cannot use modules and common blocks as an argument to the **dprint** command. (4046: 6.0.0-0)

### *Evaluation system: casting return values*

When calling functions in an Evaluation window, TotalView displays the results as a long. You can change this display by using a cast. (6.0.0-0)

### *Evaluation system: fortran intrinsics such as WRITE and COS not supported*

TotalView does not support Fortran intrinsics such as **write**, **sin**, **cos**, and **tan** in its evaluation system; i.e., in watchpoints, eval points, in the Evaluation Window.

(3842)

### *Evaluation System: Spaces required and case sensitive*

TotalView assumes that you will use spaces and lowercase when entering statements in its evaluation system; i.e., in watchpoints, evaluation points, and in the Evaluation Window. For example, the following fails:

```
if(i.ge.10.and. i.le.20) $stop
```

This will also fail:

```
if(i.ge.10 .and. i.le.20) $STOP
```

This succeeds:

```
if(i.ge.10 .and. i.le.20) $stop
```

(1985)

### *Evaluation point with a goto and a step*

If an evaluation point executes a **goto** statement or an assembly language transfer of control instruction, and you use the **step** or **next** command at the line where the evaluation point is enabled, TotalView continues the program and the **step** or **next** command does not complete. To regain control, type Ctrl+C into the program window.

### *Fortran 90 modules not supported in the evaluation system*

Fortran 90 modules are not supported by the evaluation system. On AIX, TotalView may hang when evaluating an expression containing a module variable. (4002: 6.0.0-0)

### *Fortran arrays whose size changes*

When a Variable Window displays a single element of a Fortran array that has runtime bounds (that is, assumed shape, assumed size, allocatable, or a pointer), and the actual bounds change, the value displayed in the Variable Window applies to the wrong element in the reshaped array.

You will only see this problem when all of the following conditions occur:

- The size of your Fortran array changes
- You use the **View > Variable** command
- You are only displaying a single element, either because you have dived, or because you had used an array index with a command that displays this window.

To overcome this problem, display the whole array, then dive to the element that you want to see. Alternately, if you select the specific element of interest by setting the slice expression rather than by diving, the correct element always displays, even if the array changes shape.

### *Function static variables may be invisible when using KCC*

The KCC compiler moves a static variable from the function in which it is declared and places the declaration at file or global scope. It also mangles the name to show that the variable ought to be at function scope.

### *GUI: Accelerators are overriding menu mnemonics if F10 is pressed*

TotalView's use of single letter accelerators such as **f** and **v** override the menu mnemonics (the underlined letters) when the menubar is selected using the F10 key. (2756)

### *Multithreaded corefile display problem*

TotalView shows the wrong initial thread in a multithreaded corefile. It should display the file that received the signal. (3267)

### *Portland Group Compilers (or PGI)*

TotalView 6.0 does not support any version of the **pgf90** compiler. (6.0.0-0)

### *PVM*

We have not yet certified PVM support on TotalView 6.0. Please report any problems you find to [support@etnus.com](mailto:support@etnus.com). (6.0.0-0)

### *Type transformations system revised; older version is incompatible*

The type transformation system that was previously available in TotalView 5.0 is not supported in TotalView 6.0. The new system is described in the CREATING TYPE TRANSFORMATIONS GUIDE. Go to [http://www.etnus.com/Support/docs/type\\_transformations.html](http://www.etnus.com/Support/docs/type_transformations.html) for a list of which compilers are supported. (3881: 6.0.0-0)

### *Variables in subblocks may not be found when using GCC 3.2*

There is a known compiler problem in GCC 3.2 that emits incorrect debug information for variables instantiated in subblocks; for example, for a variable whose scope is a conditional block or loop. The effect is that you may not be able to dive on these variables in TotalView 6.0. The problem also exists when using GDB and is being reported to the GCC maintainers. (4399: 6.0.0-0)

### *Watchpoints not allowed on registers*

TotalView does not support setting watchpoints on registers. If you set a watchpoint on a register variable TotalView does not display an error message. When the value in the register changes, TotalView does display an error message, but this message does not indicate what the real problem is. (3075)

### *Software and Motif problems*

If you set Software version 8 to emulate setting Motif properties, modal dialog boxes can become system modal; that is, they prevent all other window input to any window until the window is dispatched. If some other problem occurs at this time, you will need to reboot your NT workstation.

You can avoid this problem by selecting the **Windows Option** tab from within the **Options>Configuration** dialog box and set **Motif Properties** to off.

Motif text accelerators sometimes do not work. (2840)

Typing an Escape while navigating in the menus may crash TotalView if your focus policy is "X". (2714)

When displaying Help, Software often mangles how it displays pictures.

## HP Alpha Tru64 Problems

The following are problems that we know about. More details about each item follow this list.

- Anonymous unions using GNU
- Compiling with -C to detect subscripts
- Fortran 90 modules not in stack frame in HP (Compaq) Fortran (6.0.0-0)
- Opaque type showing up on Tru64 (6.0.0-0)
- Planting too many action points causes problems
- Pointers may show incorrect values using GCC 3.1.1 on Tru64 4.0f (6.0.0-0)
- prun problem finding executable file during MPI debugging. (6.0.0)
- Setting a breakpoint in a large shared-memory target causes a SEGV
- Thread debugging problems on all versions of HP Tru64

### *Anonymous unions using GNU*

The GNU compiler does not output debugging information for members of anonymous unions that are enclosed in other aggregates when using the ECOFF format on the HP Alpha. As a result, if you are debugging in such an environment, you will not see these kind of members while looking at a data structure that contains them. Furthermore, the debugging information for the offsets of aggregate members that follow the anonymous union is output incorrectly, so these members are displayed with incorrect values.



### *Compiling with -C to detect subscripts*

Compiling with -C to detect subscript out of range errors at run-time may cause TotalView to jump to a dimension statement while stepping.

On occasion, you will see a box appearing around the line number of a dimension statement when you are stepping through a routine. Due to a compiler bug in HP Alpha Fortran, several assembler instructions generated for certain bounds checks are associated with the line number of the array dimension statement rather than with the line number where the subscript is checked. When you single step a bounds-checked line and if that line contains the lowest instruction addresses associated with a dimension statement, TotalView will stop at the dimension statement. TotalView steps over other bounds-checked lines properly.

### *Fortran 90 modules not in stack frame in HP (Compaq) Fortran*

Some versions of the HP Alpha Fortran compiler do not provide information about where modules are USEd. These compilers only provide module information at the global level.

You need to specify when a module is USEd in main or in a subroutine so that you access those variables from that context within TotalView. Until this is resolved, you can get to the variables that are members of modules using **Tools > Fortran Modules** command. (3709, Echelon ticket #'s 1412 and 4742: 6.0.0-0)

### *Opaque type showing up on Tru64*

Compaq C++ V6.5-006 for Compaq Tru64 UNIX V5.1 (Rev. 732) doesn't define some variables inside STL containers and as a result Totalview shows their type as opaque type. (4408: 6.0.0-0)

### *Planting too many action points causes problems*

On a Tru64 system, using one or more TotalView commands that plant a lot of breakpoints results in an error message being displayed when you run, continue, step, or otherwise cause your program to continue or start execution.

HP is aware of the problem, but a fix is not yet available.

You can temporarily workaround this problem by using **dbx** to increase the **vt\_maxentries** variable to something like 20,000. For example:

```
dbx -k /vmunix
assign vm_tune.vt_mapentries=20000
quit
```

You can also alter **vt\_mapentries** using the **sysconfigdb** program. Consult the man page for more information.

### *Pointers may show incorrect values using GCC 3.1.1 on Tru64 4.0f*

Incorrect values may be shown by TotalView when debugging targets compiled with GCC 3.1.1 on Tru64 4.0f. The problem does not appear when using Tru64 5.1.

(4439, 4124: 6.0.0-0)

### *prun problem finding executable file during MPI debugging.*

This is a problem that has been logged against **prun**. When **prun** starts, it may receive an executable's full path. After acquiring the processes **prun** strips off the path name portion of the executable, and hands TotalView only the executable name. If TotalView's search paths are not set correctly, TotalView may be unable to find, or may find the wrong version of the program to debug. Updating to RMS 2.80-5 or later should resolve the issue.

(4275: 6.0.0-0)

### *Setting a breakpoint in a large shared-memory target causes a SEGV*

If setting a breakpoint causes the operating system to allocate shared page tables, reading information from these pages can lead to the program getting a SEGV and TotalView exiting with a **resources lost** message. You can avoid this problem by setting the value of **ssm-threshold** to 0. For example:

```
#sysconfig -r ipc ssm-threshold=0
ssm-threshold: reconfigured
#sysconfig -q ipc ssm-threshold
ipc:
ssm-threshold = 0
```

Setting this value to 0 can degrade performance.

This problem has been reported, but a fix is not yet available.

### *Thread debugging problems on all versions of HP Tru64*

Because of a bug in the Alpha thread debugging support on HP Tru64, the low-level thread hold operation can allow a held thread to run. TotalView uses the low-level

thread hold operation to prevent a thread from running when single-stepping another thread.

For example, assume your program has two threads, thread A and thread B. Assume that thread A is stopped at a breakpoint, and thread B is stopped elsewhere but not at a breakpoint. To continue the process (that is, both threads), TotalView must step thread A off the breakpoint. To do this, TotalView holds thread B. Then it unplants the breakpoint where thread A is stopped, sets a temporary breakpoint at the next instruction, and continues the process. Because of the hold thread bug, both thread A and thread B may run even though thread B is held. This means that thread B may miss the real breakpoint and hit the temporary breakpoint instead.

The following behaviors can indicate the presence of this bug:

- Threads miss breakpoints.
- Threads do not evaluate interpreted breakpoints.
- You see undeserved process stops, that is, the process may stop unexpectedly after evaluating an interpreted breakpoint.
- During single-stepping operations, threads other than the thread being stepped run.

## HP Alpha Linux Problems

The following is a problem that we know about.

### *Fortran 90 modules problem*

HP (formerly Compaq) Fortran 90 for Alpha Linux provides no Fortran 90 modules debug information. Module functions and variables are inaccessible in programs generated with this compiler.

## IBM AIX RS/6000 Problems

The following are problems that we know about. More details about each item follow this list. Items new with this release are in bold.

- AIX may only create a partial core file
- Array statistics and visualization problems when using very large arrays (6.0.0-0)
- Calling dynamic objects from Evaluation Window

- Certain XLF compiler options may show incorrect addresses for common blocks in 64-bit mode (6.0.0-0)
- Continuing from a breakpoint in a 64-bit multithreaded application may cause the application to fail (6.0.0-0)
- Multithreaded problems
- poe interferes with a standalone CLI's use of stdin
- ptrace attaching fails
- Process contention scope not supported
- pthread\_db\_pthread() returns an empty pthread list
- Signals are not delivered to the thread the user requested (6.0.0)
- Watchpoints are not supported on the Power4 architecture (6.1.0-2)
- XL Fortran: Compiler omits information about modules in OpenMP programs (6.0.0)
- XL Fortran: Not all versions supported (6.2.0-0)
- xlf 8.1.0.0 compiler emits broken module debug information (6.1.0-3)

### *AIX may only create a partial core file*

By default, AIX only dumps a partial core file. In general, a partial core dump contains only enough information to give a stack backtrace for the faulting thread. User data sections as well as some other potentially useful information are only available in a full core dump.

To force a full core dump on AIX, you must set a signal flag with **sigaction** for the signal that caused the core dump. For example:

```
struct sigaction act;
act.sa_handler = SIG_DFL;
if (bigcore)
    act.sa_flags = SA_FULLDUMP;
else if (smallcore)
    act.sa_flags = SA_PARTDUMP;
sigaction (SIGSEGV, &act, 0);
```

### *Array statistics and visualization problems when using very large arrays*

Using array statistics and array visualization on some large Fortran arrays with TotalView for IBM AIX can cause memory consumption problems. If this occurs, TotalView may appear to hang or may exit with a memory allocation fatal error. (3870, 3872, Echelon ticket# 5478: 6.0.0-0)

### *Calling dynamic objects from Evaluation Window*

If a routine in a dynamic object is called from the expression window, and if the target routine is never called from the main program, TotalView refuses to call the routine. (6.0.0-0)

### *Certain XLF compiler options may show incorrect addresses for common blocks in 64-bit mode*

When compiling an OpenMP program, the combination of the **-qsmp** and **-q64** compiler options appear to generate incorrect debug information for the location of large common blocks. When diving onto the common block, TotalView will show incorrect values since the debug information is pointing it to the wrong starting location for the common.

### *Continuing from a breakpoint in a 64-bit multithreaded application may cause the application to fail*

This is an AIX 5.1 kernel bug. The bit that indicates 32-bit/64-bit application code in the MSR (Machine Status Register) is cleared during one of the system calls. This can clear the high order 32 bits in the 64-bit address under some circumstances when trying to continue from the breakpoint. Your program may subsequently seg fault or get an illegal instruction trap.

Contact Etnus customer support for instructions on obtaining a fix from IBM if you encounter this problem. You may also contact IBM technical support directly. They are tracking the issue as APAR IY36607. (4216: 6.0.0-0)

### *Multithreaded problems*

You may experience some problems when debugging multithreaded programs, because of limitations in the **ptrace()** operating system call.

The following problems can show up while you are debugging multithreaded applications:

- 1 When a thread stops (for example, hits a breakpoint) all the other threads stop. If any of the other threads stops while in a system call (for example, **read()**, **sleep()**, **select()**, etc.), however, **ptrace()** does not allow the debugger to read the thread's registers. As a result, TotalView:
  - Cannot display the registers, including the program counter; but does display the stack pointer
  - Cannot show you which system call is being executed

- Cannot single-step using the *step* or *next* command, but *out* and *run to work*
- Cannot display the top stack frame

If you have a multithreaded application that makes a lot of system calls, it is possible that most of your threads are not fully debuggable whenever one of them stops.

- 2 TotalView shows you which threads are stuck in the kernel by displaying their state as In Kernel (**K**).
- 3 When a thread is created or destroyed, the system does not notify the debugger of this event. As a result, the list of threads displayed by TotalView may be stale when the program is running.
- 4 If the process stops for any reason, TotalView automatically updates the thread list. You may also type **View > Reset** to force the thread list to update.

#### *poe interferes with a standalone CLI's use of stdin*

Because **poe** tries to manage **stdin** on behalf of its target processes, a CLI invoked directly from the shell cannot read from **stdin**. If your target processes do not use **stdin**, using the **-stdinmode none** option to the **poe** command allows the CLI to use **stdin**. Unfortunately, this option is incompatible with the **poe** command's **-cmdfile** option.

If your processes do use **stdin**, your only recourse is to redirect **stdin** from within the CLI. For example:

```
drun < in.txt
```

(pr 2078, 2422)

#### *ptrace attaching fails*

Versions of the AIX kernel after AIX 4.3.3.1 contain a bug that causes a **ptrace()** attach to fail for some programs. In particular, attaching to a Parallel Environment program may fail. You can solve this problem by installing IY10784, whose description is:

**IY10784: ATTACH FAILS TO THE CHILD PROCESS OF A ROOT PROCESS**

#### *Process contention scope not supported*

On AIX systems, TotalView supports debugging pthread programs running in pthread-compatibility mode or pthreads scheduled in system contention scope, that is, each pthread is bound to a kernel thread (the 1:1 thread scheduling model).

TotalView does not support process contention scope, that is, multiple pthreads scheduled in user mode (M:N thread scheduling model).

On AIX, when using TotalView to debug a program built with **libpthreads.a**, you must force the 1:1 model using the procedure outlined in **Forcing 1:1 Thread Scheduling Mode on RS/6000 Systems**, which is contained in the IBM SPECIAL CONSIDERATIONS document.

### *pthdb\_pthread() returns an empty pthread list*

Sometimes when a process is stopped and the **pthdb\_pthread()** function is used to obtain a list of pthreads, the returned list is empty even when there are pthreads. (TotalView displays a console message saying that there are no more threads.) You can fix this problem by applying the APAR IY06378 patch to your system. The procedure for obtaining and applying patches is described in **RS/6000 System Patch Procedures**, which is found in the IBM SPECIAL CONSIDERATIONS document.

### *Signals are not delivered to the thread the user requested*

**ptrace()** only allows TotalView to deliver a signal to the primary thread that is being continued, and that thread must be the *distinguished* thread ; that is, it is the one that caused the last exception.

If you tell TotalView to continue a particular thread with a signal, TotalView can only send it to the current distinguished thread. Most of the time, this means the thread you want to signal will not be the one that will actually receives it. This is true regardless of whether the distinguished thread will also be run or if it is being held.

Note that this does not affect reforwarding of signals that were originally received by the process (for example, for a user SEGV handler). Those are also sent to the distinguished thread, but since it's the thread that the signal stopped, it's the right thread to receive the signal.

### *Watchpoints are not supported on the Power4 architecture*

Watchpoint support is not yet provided by AIX for Power4 processors. When planting a watchpoint, TotalView displays the following error message in the console: (6.1.0-2)

Error planting action point at address 0x##### length #

***XL Fortran: Compiler omits information about modules in OpenMP programs***

Problems have been seen looking at module data in XL Fortran compiled OpenMP programs. Users experiencing this problem will note that modules are missing from the Program Browser Window. PR3769 (Echelon ticket# 4046: 6.0.0-0)

***XL Fortran: Not all versions supported***

Versions of IBM XL Fortran 8.1 prior to 8.1.0.3 should not be used with TotalView. You must update to XL Fortran 8.1.0.3. (6.2.0-2)

***xlf 8.1.0.0 compiler emits broken module debug information***

To fix this problem, you should, at a minimum, apply the latest patch to the compiler, 08.01.0000.0003 (xlf 8.1 PTF3). (6.1.0-3)

**Linux ia64 Problems**

The following is a problem that we know about.

***SGI's MPT not supported***

SGI's MPT for Altix is not supported by TV 6.2.0-0. (4761)

**Linux x86 Problems**

The following are problems that we know about. More details about each item follow this list.

- Breakpoints in C++ constructors in shared libraries problem (6.0.0-0)
- Calling exec() from a thread problems
- GCC g77 problem with common blocks
- GLIBC update required for RedHat 7.3 (6.0.0-0)
- Intel Fortran: 128-bit real support issue (6.1.0-2)
- Licensing problem using Intel Hyperthreading Processors (6.0.0-1)
- Multithreaded corefiles on Linux are not understood. (6.0.0-0)
- Native Posix Thread Library (NPTL) not yet supported (6.2.0-0)
- Opening message queue on some versions of Quadrics/RMS crashes TotalView (6.0.0-0)
- PGI Problems (6.0.0-0)
- Red Hat Linux kernel 2.4 security update causes seg fault (6.2.0-0)
- Red Hat 9 not yet supported (6.2.0-0)



- Stepping into system routine backtrace problem (5.0.0-0)
- Thread debugging and errno

### ***Breakpoints in C++ constructors in shared libraries problem***

Static constructors in shared libraries that execute as part of the library initialization are sometimes executed before TotalView plants breakpoints. Breakpoints in this code will not work as expected. (3801, Echelon ticket# 5283: 6.0.0-0)

### ***Calling exec() from a thread problems***

Debugging threaded programs (pthreads) that call **exec()** from a thread is not yet supported.

### ***GCC g77 problem with common blocks***

The GCC g77 compilers do not output debugging information for common blocks. Consequently, TotalView cannot show the values of variables in common blocks.

### ***GLIBC update required for RedHat 7.3***

The **glibc** package must be updated to at least version 2.2.5-40 in order for TotalView 6.0 to work properly on RedHat 7.3. RPM packages are available from RedHat at the following URLs.

<ftp://updates.redhat.com/7.3/en/os/i386/glibc-2.2.5-40.i386.rpm>

<ftp://updates.redhat.com/7.3/en/os/i386/glibc-common-2.2.5-40.i386.rpm>

(6.0.0-0)

### ***Intel Fortran: 128-bit real support issue***

The **real(kind=16)** and **complex(kind=32)** types are only partially supported. When displayed in a decimal format, the values are truncated to **real(kind=8)** and **complex(kind=16)** before they are displayed. Similarly, if the value is updated using a decimal format, the low significance bits are set incorrectly. (6.1.0-2)

### ***Licensing problem using Intel Hyperthreading Processors***

Recent Intel IA32 processors may be used in a mode called *Hyperthreading*. In this mode, the Linux kernel presents the image of two CPUs for each physical CPU to programs running on the system. TotalView relies on the Linux kernel to obtain CPU count information. The kernel supplied with RedHat versions up to and including 8.0 does not allow TotalView to get a correct CPU count. Etnus has obtained a

kernel patch from Intel that can be easily applied. Contact [support@etnus.com](mailto:support@etnus.com) to obtain this patch. (6.0.0-1)

#### ***Multithreaded corefiles on Linux are not understood.***

The RedHat Linux kernels against which TotalView is built and tested do not generate multithreaded corefiles. It is unlikely that TotalView would correctly interpret a multithreaded corefile generated by a Linux kernel that had been compiled with a patch to provide multithreaded corefiles. (6.0.0-0)

#### ***Native Posix Thread Library (NPTL) not yet supported***

The Native Posix Thread Library (NPTL) threads implementation for Linux is not yet supported. (4613: 6.2.0-0)

#### ***Opening message queue on some versions of Quadrics/RMS crashes TotalView***

This is due to a problem with ptrace accessing memory that is mapped to the ELAN. Users should update to OSNETLIBS 1.4.3.1. (4283, Echelon ticket# 6130: 6.0.0-0)

#### ***PGI Problems***

PGI compilers are not supported in TotalView 6.0. (6.0.0-0)

#### ***Red Hat Linux kernel 2.4 security update causes seg fault***

The Red Hat Linux kernel 2.4 security update fix for ptrace vulnerability causes TotalView to crash with an immediate segmentation fault.

A kernel update RPM package was released for RedHat 7.1, 7.2, 7.3, and 8.0 that addresses a vulnerability in ptrace. This fix introduced a new bug in the kernel that causes TotalView to crash and introduces other kernel stability problems. The kernel developers have located problem and a new fix should be forthcoming.

If you are experiencing this problem and build your own Linux kernels, the solution is to change the definition of `is_dumpable` in `include/linux/sched.h` from:

```
#define is_dumpable(tsk)((tsk)->task_dumpable \
    && (tsk)->mm->dumpable)
```

to:

```
#define is_dumpable(tsk)((tsk)->task_dumpable \
    && (tsk)->mm && (tsk)->mm->dumpable)
```

The following affected Red Hat RPM packages should not be installed:

### Red Hat Linux 7.1

- kernel-2.4.18-27.7.x.src.rpm
- kernel-2.4.18-27.7.x.athlon.rpm
- kernel-smp-2.4.18-27.7.x.athlon.rpm
- kernel-2.4.18-27.7.x.i386.rpm
- kernel-BOOT-2.4.18-27.7.x.i386.rpm
- kernel-doc-2.4.18-27.7.x.i386.rpm
- kernel-source-2.4.18-27.7.x.i386.rpm
- kernel-2.4.18-27.7.x.i586.rpm
- kernel-smp-2.4.18-27.7.x.i586.rpm
- kernel-2.4.18-27.7.x.i686.rpm
- kernel-bigmeme-2.4.18-27.7.x.i686.rpm
- kernel-debug-2.4.18-27.7.x.i686.rpm
- kernel-smp-2.4.18-27.7.x.i686.rpm

### Red Hat Linux 7.2

- kernel-2.4.18-27.7.x.src.rpm
- kernel-2.4.18-27.7.x.athlon.rpm
- kernel-smp-2.4.18-27.7.x.athlon.rpm
- kernel-2.4.18-27.7.x.i386.rpm
- kernel-BOOT-2.4.18-27.7.x.i386.rpm
- kernel-doc-2.4.18-27.7.x.i386.rpm
- kernel-source-2.4.18-27.7.x.i386.rpm
- kernel-2.4.18-27.7.x.i586.rpm
- kernel-smp-2.4.18-27.7.x.i586.rpm
- kernel-2.4.18-27.7.x.i686.rpm
- kernel-bigmeme-2.4.18-27.7.x.i686.rpm
- kernel-debug-2.4.18-27.7.x.i686.rpm
- kernel-smp-2.4.18-27.7.x.i686.rpm

### Red Hat Linux 7.3

- kernel-2.4.18-27.7.x.src.rpm
- kernel-2.4.18-27.7.x.athlon.rpm
- kernel-smp-2.4.18-27.7.x.athlon.rpm

- kernel-2.4.18-27.7.x.i386.rpm
- kernel-BOOT-2.4.18-27.7.x.i386.rpm
- kernel-doc-2.4.18-27.7.x.i386.rpm
- kernel-source-2.4.18-27.7.x.i386.rpm
- kernel-2.4.18-27.7.x.i586.rpm
- kernel-smp-2.4.18-27.7.x.i586.rpm
- kernel-2.4.18-27.7.x.i686.rpm
- kernel-bigmem-2.4.18-27.7.x.i686.rpm
- kernel-debug-2.4.18-27.7.x.i686.rpm
- kernel-smp-2.4.18-27.7.x.i686.rpm

### Red Hat Linux 8.0

- kernel-2.4.18-27.8.0.src.rpm
- kernel-2.4.18-27.8.0.athlon.rpm
- kernel-smp-2.4.18-27.8.0.athlon.rpm
- kernel-2.4.18-27.8.0.i386.rpm
- kernel-BOOT-2.4.18-27.8.0.i386.rpm
- kernel-doc-2.4.18-27.8.0.i386.rpm
- kernel-source-2.4.18-27.8.0.i386.rpm
- kernel-2.4.18-27.8.0.i586.rpm
- kernel-smp-2.4.18-27.8.0.i586.rpm
- kernel-2.4.18-27.8.0.i686.rpm
- kernel-bigmem-2.4.18-27.8.0.i686.rpm
- kernel-debug-2.4.18-27.8.0.i686.rpm
- kernel-smp-2.4.18-27.8.0.i686.rpm

### *Red Hat 9 not yet supported*

Red Hat 9 is not yet supported since it includes the Native Posix Thread Library (NPTL). (4762: 6.2.0-0)

### *Stepping into system routine backtrace problem*

If you step into a system routine such as **printf()**, TotalView may stop in the dynamic linker before its information has been fixed up. In this case, you may not be able to return out of this place.

You can avoid this problem by setting **LD\_BIND\_NOW** to 1 (see Chapter 9 of the TOTALVIEW REFERENCE GUIDE for more information) or you can use the **Next** command instead of the **Step** command. (3015)

### *Thread debugging and errno*

When using pthreads on Linux, the **errno** variable is actually a C macro defined as follows in **bits/errno.h**:

```
#define errno (*__errno_location ())
```

This definition allows each thread to have its own **errno** value. Unfortunately, the program does not contain information that allows TotalView to find this thread specific **errno** value and there remains a single global **errno** variable still exists.

The result is that displaying **errno** in any thread other than the initial one in a process is likely to be very misleading, since you will see the global **errno** variable, rather than the per-thread value accessed by your code through the above macro.

## SGI IRIX Problems

The following are problems that we know about. More details about each item follow this list. Items new with this release are in bold.

- Arrays in main are not found unless declared in common
- Cray pointers in common blocks broken
- Evaluation system forces real function result into a long temporary
- #include and -cpp Used Together in Fortran 90
- KCC does not put original file name into symbol table
- main not found by TotalView
- Values in assumed sized arrays may be wrong for F77 compiled routines (6.0.0-1)

### *Arrays in main are not found unless declared in common*

If an array is declared in **main**, the SGI MIPSpro 7.3.3 compiler does not create debugging information for the variable. Consequently, TotalView does not know that the array exists. You can workaround this problem by placing the array in a common block.

### *Cray pointers in common blocks broken*

The debugging information generated by SGI 7.3 Fortran compiler for the targets of Cray pointers contained within common blocks contains the wrong address. Here is an example:

```
common a1(1000)
common /ptrs/ jj,iparray,kk
pointer (iparray,array)
iparray = loc(a1)
end
```

**array** is a real variable that is the target of the Cray pointer **iparray**. Because the address is wrong, TotalView cannot show you the correct values for the **iparray** variable. This bug has been reported to SGI.

### *Evaluation system forces real function result into a long temporary*

When a program is compiled with the Fortran 90 compiler, the TotalView expression evaluation system erroneously converts real function results. The SGI Fortran 90 compiler fails to emit the return type of the function, so TotalView assumes that the return type of the function is a long. When assigned to a real variable, the return result of the function is erroneously converted from a long to a real, when in fact the function had already returned a real. Here is an example:

```
real function x_to_the_y_power(x, y)
```

TotalView expression:

```
real result
result = x_to_the_y_power(2.0, 4.0)
```

This problem, which does not occur with the Fortran 77 compiler, has been reported to SGI. (pr 2296)

### *#include and -cpp Used Together in Fortran 90*

If source files contain **#include** statements and are compiled with the **-cpp** switch on a Fortran 90 program using the MIPSpro compilers, TotalView generates incorrect line numbers. To avoid this problem, use the standard Fortran **include** statement (without the **-cpp** switch).

***KCC does not put original file name into symbol table***

IRIX KCC code: TotalView fails to put the original file name (before preprocessing) into the symbol table. This prevents you from asking for the file by name until TotalView processes all the file's symbols.

If you use the `--keep_gen_c` option to the KCC compiler, you can use the following TotalView command: `f xxx.int.c` (where your original source file was `xxx.C`) to force full symbol processing of that file, after which you'll be able to do `f xxx.C`.

***main not found by TotalView***

TotalView will not find a Fortran 90 **main** program. TotalView will not display any source code if you do not use a **PROGRAM** statement within a Fortran 90 program. You can correct this problem by adding a **PROGRAM** statement to your **main** program. (2099)

***Values in assumed sized arrays may be wrong for F77 compiled routines***

TotalView may show incorrect array values for assumed sized arrays when using the SGI MIPSpro 7.30 F77 compiler. This is due to a problem in the debug information generated by the compiler. To work around the issue, use the SGI MIPSpro 7.30 F90 compiler or later versions of the F77 compiler. (4513)

**Sun SPARC Solaris Problems**

The following are problems that we know about. More details about each item follow this list. Items new with this release are in bold.

- Apogee 4.0 compilers must be patched
- Breakpoints in thunks may cause crash
- SUNPERF runs slow when debugging a Fortran 90 program (6.1.0-2)

***Apogee 4.0 compilers must be patched***

The Apogee 4.0 compilers on Sun OS4 and Sun OS5 require a patch to bring them up to revision level 4.013. Follow the **Apogee 4.0 Compiler Patch Procedures**, which is found in the **Patching** document.

***Breakpoints in thunks may cause crash***

Using breakpoints in thunks may lead to unexpected results, including having the target program crash unexpectedly. A thunk is a small linkage routine that connects a subroutine call to the actual subroutines in a dynamic library. The SPARC Solaris

dynamic loader modifies the code in the thunks during program execution, which conflicts with TotalView's planting and unplanting of breakpoints. The first time through a thunk, the thunk branches to the dynamic loader, and the dynamic loader modifies the thunk to branch directly to the corresponding dynamic library routine. Subsequent trips through the thunk branch directly to the dynamic library routine.

### *SUNPERF runs slow when debugging a Fortran 90 program*

The SUNPERF library from Sun is stripped of debug information. When using F90 modules from this library in your F90 application, the compiler inserts debug information into the executable that references the symbols that were removed from this library. Unfortunately, TotalView is not able to determine that the symbols inserted by the compiler will be unresolvable, which causes debugging these targets to be slow. (4584)

## Reporting Problems

If you experience any problems with TotalView, or if you have questions or suggestions, please contact us:

Etnus LLC.

24 Prime Parkway

Natick, MA 01760

Email: [support@etnus.com](mailto:support@etnus.com)

Phone: 1-800-856-3766 in the United States

(+1) 508-652-7700 worldwide

## Notices

Copyright (c) 1999-2002 by Etnus LLC. All rights reserved

Copyright (c) 1999 by Etnus Inc. All rights reserved

Copyright (c) 1996-1998 by Dolphin Interconnect Solutions, Inc.

Copyright (c) 1993-1996 by BBN Systems and Technologies, a division of BBN Corporation.



All other brand names are the trademarks of their respective holders.



# Index

## Symbols

- #include and -cpp Used Together in Fortran 90 38
- .tvdrc File 2
- "Attempt to get an address from a located symbol ..." problem fixed 16
- "Current scope not a block" error fixed 17
- "Duplicate entry in table" error 17
- "exAssert in file tvpanes.cxx line 1190" problem fixed 9

## Numerics

- 128-bit real support issue
  - Intel Fortran 33
- 2 GB code size problem fixed 11
- 9 not yet supported
  - Red Hat 36
- 90 breakpoint problem fixed
  - Fortran 17
- 90 modules not in stack frame in HP (Compaq) Fortran
  - Fortran 25
- 90 modules not supported in the evaluation system
  - Fortran 22
- 90 modules problem
  - Fortran 27

## A

- a breakpoint file for a shared library could crash TotalView
  - Loading 6, 15

- a breakpoint in a large shared-memory target causes a SEGV
  - Setting 26
- a Fortran 90 local variable problem fixed
  - Diving on 11
- Accelerators are overriding menu mnemonics if F10 is pressed
  - GUI 23
- Action Point > At Location
  - No longer torn down 4
  - Typing C++ member function 4
- AIX may only create a partial core file 28
- Ambiguous functions from dynamically loaded shared libraries are found when looking up a function 5
- an Fortran 90 variable can cause a fatal error (IRIX and Linux)
  - Diving on 5
- and erno
  - Thread debugging 37
- Anonymous unions using GNU 24
- Apogee 4.0 compilers must be patched 39
- Apparent hanging problem fixed
  - CodeRoad JNI Bridge 17
- are not supported on the Power4 architecture
  - Watchpoints 31

- Array statistics and visualization problems when using very large arrays 28
- Arrays in main are not found unless declared in common 37
- arrays whose size changes
  - Fortran 22
- Assembly shown by mistake
  - Lahey/Fujitsu Fortran 14
- Attaching problem fixed 13
- attaching problem fixed
  - CodeRoad JNI Bridge 13

## B

- Breakpoints
  - in C++ constructors in shared libraries problem 33
  - in thunks may cause crash 39
  - no longer disappear when using group step, next, or run to 15
- breakpoints in functions located in two different libraries problem fixed
  - Setting 7
- bulk launch no longer crashes tvdsvr
  - prun 10
- Bulk launch sometimes failed if single debug server launch was disabled 5
- Buttons now responsive if two process windows are open 5

**C**

- C++
  - class variables were incorrectly marked as 'Stale' 5
  - exceptions 19
  - not supported in the evaluation system 20
  - static variables not visible 9
- Calling
  - dynamic objects from Evaluation Window 29
  - exec() from a thread problems 33
- Cannot be used until process starts
  - Program Browser 6
- casting return values
  - Evaluation system 21
- causing seg fault problem fixed
  - Reading symbols 10
- Certain XLF compiler options may show incorrect addresses for common blocks in 64-bit mode 29
- class variables were incorrectly marked as 'Stale'
  - C++ 5
- Cluster Tools 5 MPI problem fixed
  - Sun 17
- CodeRoad JNI Bridge
  - Apparent hanging problem fixed 17
  - attaching problem fixed 13
- Compiler omits information about modules in OpenMP programs
  - XL Fortran 32
- Compiling with -C to detect subscripts 25
- configure\_autostart licensing problem fixed 13
- Continuing from a breakpoint in a 64-bit multithreaded application may cause the application to fail 29
- corefile display problem
  - Multithreaded 23

- corefiles on Linux are not understood.
  - Multithreaded 34
- crash problem fixed
  - TotalView 16
- Cray pointers
  - in common blocks broken 38
  - issue 11
- cset demangler option replaced by xlf 3

**D**

- DBX class tag errors 17
- Defaults Command No Longer Crashes TotalView
  - File > Preferences > Formatting 14
- Delete problem fixed
  - Group > 6
- Diving on
  - a Fortran 90 local variable problem fixed 11
  - an Fortran 90 variable can cause a fatal error (IRIX and Linux) 5
- dprint does not display the contents of a common block or module. 21
- dynamic objects from Evaluation Window
  - Calling 29

**E**

- Error Message Problem
  - resolve\_final\_symbol error 5
- Evaluation point with a goto and a step 22
- Evaluation System
  - Spaces required and case sensitive 21
- Evaluation system
  - casting return values 21
  - forces real function result into a long temporary 38
  - fortran intrinsics such as WRITE and COS not supported 21
- exceptions
  - C++ 19
- exec() from a thread problems
  - Calling 33
- executable problem fixed
  - Loading 9

**F**

- Fatal Error starting up with very large search path values
  - Search Paths 7
- File > Preferences > Formatting
  - Defaults Command No Longer Crashes TotalView 14
- Files can now be found through symbolic links 9
- Fixed
  - Problems 3
  - SGI IRIX Problems 15
- Fixed On All Platforms
  - Problems 3
- Focus problem
  - MPI and Sun ClusterTools 5
  - 17
- for pointer problem fixed
  - Reading symbols 10
- forces real function result into a long temporary
  - Evaluation system 38
- Form\_long result too large problem fixed 11
- Fortran
  - 90 breakpoint problem fixed 17
  - 90 modules not in stack frame in HP (Compaq) Fortran 25
  - 90 modules not supported in the evaluation system 22
  - 90 modules problem 27
  - arrays whose size changes 22
  - logical type issue 11
  - modules list item order 5
  - Modules Support 15
  - parameter lookup problem fixed 11

- static link problem 12
- fortran intrinsics such as WRITE and COS not supported
  - Evaluation system 21
- Function static variables may be invisible when using KCC 22

## G

- g77 problem with common blocks
  - GCC 33
- GCC
  - g77 problem with common blocks 33
  - gstabs compiler option support is improved 14
- GCC 3.1
  - TotalView can parse debug information For stl::deque 9
- GCC 3.1 and 3.2
  - Opaque Type Problem When Base Class in a Different Shared Library Problem Resolved 14
- GCC 3.2
  - Internal Error starting up upon a GCC 3.2 compiled executable 11
- GCC 3.2 g77
  - General note 6
- General note
  - GCC 3.2 g77 6
- GLIBC update required for RedHat 7.3 33
- Group >
  - Delete problem fixed 6
  - Reload symbols command no longer necessary 2
- gstabs compiler option support is improved
  - GCC 14
- GUI
  - Accelerators are overriding menu mnemonics if F10 is pressed 23

## H

- HP Alpha
  - Linux Problems 27
  - Tru64 Problems 24
  - Tru64 Problems Fixed 8

## I

- ia64 Problems
  - Linux 32
- IA-64 Problems Fixed
  - Linux 12
- IBM AIX
  - Problems Fixed 10
  - RS/6000 Problems 27
- in C++ constructors in shared libraries problem
  - Breakpoints 33
- in common blocks broken
  - Cray pointers 38
- in thunks may cause crash
  - Breakpoints 39
- Incorrect statistics and visualization of UPC distributed arrays 15
- Intel Fortran
  - 128-bit real support issue 33
- Internal Error starting up upon a GCC 3.2 compiled executable GCC 3.2 11
- issue
  - Cray pointers 11

## K

- KCC
  - Startup problem resolved 9
- KCC 4.0F
  - Problem parsing information from C++ class solved 9
- KCC does not put original file name into symbol table 39
- Keyboard accelerator notation standardized 6
- Known Problems 18

## L

- Lahey/Fujitsu Fortran

- Assembly shown by mistake 14
  - Showing wrong source 14
  - Stepping problems fixed 14
- Large data segments no longer crash TotalView 11
- Launch strings now include %B by default 6
- libpthread.so problem resolved 9
- Licensing problem using Intel Hyperthreading Processors 33
- Linux
  - ia64 Problems 32
  - IA-64 Problems Fixed 12
  - x86 Problems 32
  - x86 Problems Fixed 13
- Linux kernel 2.4 security update causes seg fault
  - Red Hat 34
- Linux Problems
  - HP Alpha 27
- Loading
  - a breakpoint file for a shared library could crash TotalView 6, 15
  - executable problem fixed 9
- Local variables
  - Sun WorkShop 5.0 18
- logical type issue
  - Fortran 11

## M

- main not found by TotalView 39
- Memory Usage command now shows correct values of heap usage 9
- Message Queue Graph node placement 6
- Miscounting processors 14
- Missing library load notifications when using JNI Bridge problem fixed 17
- modules list item order
  - Fortran 5
- Modules Support
  - Fortran 15
- MPI and Sun ClusterTools 5

Focus problem 17  
 MPI processes randomly stop before hitting a user breakpoint during startup 12  
 MPT Support 12  
 Multi-dimensioned const arrays no longer crashes TotalView 12  
 Multithreaded  
   corefile display problem 23  
   corefiles on Linux are not understood. 34  
   problems 29  
 Multi-threaded core debugging problem fixed 12

## N

Native Posix Thread Library (NPTL) not yet supported 34  
 News  
   TotalView 2  
 no longer disappear when using group step, next, or run to Breakpoints 15  
 No longer torn down  
   Action Point > At Location 4  
 Not all versions supported  
   XL Fortran 32  
 not allowed on registers  
   Watchpoints 23  
 not supported in the evaluation system  
   C++ 20  
 Notices 40

## O

older version is incompatible 23  
 on All Platforms  
   Problems 18  
 Opaque Type Problem When Base Class in a Different Shared Library Problem Resolved  
   GCC 3.1 and 3.2 14  
 Opaque type showing up on Tru64 25  
 Opening message queue on some versions of Quadrics/RMS crashes TotalView 34

Optimizing compiler variables no longer show misleading values 10

## P

parameter lookup problem fixed  
   Fortran 11  
 PGI Problems 34  
   -pid command line option added 6  
 Planting too many action points causes problems 25  
 poe interferes with a standalone CLI's use of stdin 30  
 Pointers may show incorrect values using GCC 3.1.1 on Tru64 4.0f 26  
 Portland Group Compilers (or PGI) 23  
 printf() problems 14  
 problem finding executable file during MPI debugging.  
   prun 26  
 Problem parsing information from C++ class solved  
   KCC 4.0F 9  
 Problems  
   Fixed 3  
   Fixed On All Platforms 3  
   on All Platforms 18  
 problems  
   Multithreaded 29  
 Problems Fixed  
   IBM AIX 10  
 problems on all versions of HP Tru64  
   Thread debugging 26  
 Process contention scope not supported 30  
 Program Browser  
   Cannot be used until process starts 6  
 prun  
   bulk launch no longer crashes tvdsrv 10

problem finding executable file during MPI debugging. 26  
 pthdb\_pthread() returns an empty pthread list 31  
 ptrace attaching fails 30  
 PVM 23

## R

Reading symbols  
   causing seg fault problem fixed 10  
   for pointer problem fixed 10  
 Red Hat  
   9 not yet supported 36  
   Linux kernel 2.4 security update causes seg fault 34  
 Reload symbols command no longer necessary  
   Group > 2  
 Reloading breakpoint file problem fixed 10  
 Reporting Problems 40  
 resolve\_final\_symbol error  
   Error Message Problem 5  
 RMS prun and Message Queue 14  
 RS/6000 Problems  
   IBM AIX 27

## S

Search Paths  
   Fatal Error starting up with very large search path values 7  
 Setting  
   a breakpoint in a large shared-memory target causes a SEGV 26  
   breakpoints in functions located in two different libraries problem fixed 7  
 SGI IRIX Problems 37  
   Fixed 15  
 SGI's MPT not supported 32  
 Showing wrong source  
   Lahey/Fujitsu Fortran 14  
 Signals are not delivered to the thread the user requested 31

Spaces required and case sensitive  
 Evaluation System 21  
 SPARC Solaris Problems  
 Sun 39  
 SPARC Solaris Problems Fixed  
 Sun 16  
 Spell checker problems resolved  
 7  
 Startup problem resolved  
 KCC 9  
 static link problem  
 Fortran 12  
 static variables not visible  
 C++ 9  
 Stepping into system routine  
 backtrace problem 36  
 Stepping problems fixed  
 Lahey/Fujitsu Fortran 14  
 STL type transformations limitation 16  
 Sun  
 Cluster Tools 5 MPI problem  
 fixed 17  
 SPARC Solaris Problems 39  
 SPARC Solaris Problems Fixed  
 16  
 Sun WorkShop 5.0  
 Local variables 18  
 SUNPERF runs slow when debugging a Fortran 90 program 40

**T**  
 terminates with an internal error  
 and no other information

TotalView 18  
 Thread debugging  
 and errno 37  
 problems on all versions of  
 HP Tru64 26  
 Threads not proceeding 7  
 TotalView  
 crash problem fixed 16  
 News 2  
 terminates with an internal  
 error and no other information 18  
 Tru64 Problems  
 HP Alpha 24  
 Tru64 Problems Fixed  
 HP Alpha 8  
 tvpane.cxx crash 7  
 Type transformations system revised 23  
 Typing C++ member function  
 Action Point > At Location 4

**U**  
 UPC shared addresses of shared  
 variables and shared pointers  
 given as local 7  
 Using libdbfork on AIX 2

**V**  
 Value of remote elements of  
 shared arrays not correctly  
 represented 16  
 Values in assumed sized arrays  
 may be wrong for F77 compiled routines 39

Variable names in Variable Window title 7  
 Variables in subblocks may not  
 be found when using GCC 3.2  
 23  
 -verbosity command line option  
 no takes precedence 8  
 View > Lookup Function/File  
 problem 8

## W

Watchpoint warning message  
 suppressed 15  
 Watchpoints  
 are not supported on the  
 Power4 architecture 31  
 not allowed on registers 23  
 Window focus changes affected  
 some users 8

## X

X Resources and preferences 3  
 x86 Problems  
 Linux 32  
 x86 Problems Fixed  
 Linux 13  
 XL Fortran  
 Compiler omits information  
 about modules in  
 OpenMP programs 32  
 Not all versions supported 32  
 xlf 8.1.0.0 compiler emits broken  
 module debug information 32  
 Software and Motif problems 24

